



US Patent & Trademark Office

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)
Search: ☒ The ACM Digital Library ☐ The Guide**SEARCH**

THE ACM DIGITAL LIBRARY


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)
Terms used IDL dispatch

Found 144 of 147,793

Sort results by [Save results to a Binder](#)[Try an Advanced Search](#)Display results [Search Tips](#)[Try this search in The ACM Guide](#)☐ Open results in a new window

Results 1 - 20 of 144

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [next](#)Relevance scale ☐ ☐ ☐ ☐ ☐**1** [Customizing IDL mappings and ORB protocols](#)

Girish Welling, Maximilian Ott

April 2000 **IFIP/ACM International Conference on Distributed systems platforms**Full text available: [pdf\(293.12 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#)

Current mappings of IDL to implementation languages such as C++ or Java use CORBA specific data-types, which makes it imperative for an object implementation to be CORBA-compliant. While being completely CORBA-compliant ensures portability *and* interoperability, several classes of enterprise applications may *only* require interoperability with other CORBA applications. Other applications may be constrained by such factors as a large existing code-base or a widely used communicatio ...

2 [Resolving feature convolution in middleware systems](#)

Charles Zhang, Hans-Arno Jacobsen

October 2004 **ACM SIGPLAN Notices , Proceedings of the 19th annual ACM SIGPLAN Conference on Object-oriented programming, systems, languages, and applications**, Volume 39 Issue 10Full text available: [pdf\(569.29 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Middleware provides simplicity and uniformity for the development of distributed applications. However, the modularity of the architecture of middleware is starting to disintegrate and to become complicated due to the interaction of too many orthogonal concerns imposed from a wide range of application requirements. This is not due to bad design but rather due to the limitations of the conventional architectural decomposition methodologies. We introduce the principles of horizontal decompositi ...

Keywords: aspect oriented middleware, middleware architecture**3** [The design and performance of a scable ORB architecture for COBRA asynchronous messaging](#)

Alexander B. Arulanthu, Carlos O'Ryan, Douglas C. Schmidt, Michael Kircher, Jeff Parsons

April 2000 **IFIP/ACM International Conference on Distributed systems platforms**Full text available: [pdf\(174.72 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

Historically, method-oriented middleware, such as Sun RPC, DCE, Java RMI, COM, and CORBA, has provided synchronous method invocation (SMI) models to applications. Although SMI works well for conventional client/server applications, it is not well-suited for

high-performance or real-time applications due to its lack of scalability. To address this problem, the OMG has recently standardized an asynchronous method invocation (AMI) model for CORBA. AMI provides CORBA with many of the capabilit ...

4 Transport layer abstraction in event channels for embedded systems

Ravi Pratap M, Ron K. Cytron, David Sharp, Edward Pla

June 2003 **ACM SIGPLAN Notices , Proceedings of the 2003 ACM SIGPLAN conference on Language, compiler, and tool for embedded systems**, Volume 38 Issue 7

Full text available:  pdf(407.97 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)


As embedded systems increase in complexity and begin to participate in distributed systems, the need for middleware in building such systems becomes imperative. However, the use of middleware that fully implements such standards can impose a significant increase in footprint for an application, making it unsuitable for use in embedded systems. We consider the use of a standard CORBA event channel in a setting where distribution and inter-language support are unnecessary. We report our experience ...

Keywords: AOP, CORBA, embedded systems, event service, middleware, software composition, subsetting, transport abstraction

5 Flick: a flexible, optimizing IDL compiler

Eric Eide, Kevin Frei, Bryan Ford, Jay Lepreau, Gary Lindstrom

May 1997 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1997 conference on Programming language design and implementation**, Volume 32 Issue 5

Full text available:  pdf(1.75 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

An interface definition language (IDL) is a nontraditional language for describing interfaces between software components. IDL compilers generate "stubs" that provide separate communicating processes with the abstraction of local object invocation or procedure call. High-quality stub generation is essential for applications to benefit from component-based designs, whether the components reside on a single computer or on multiple networked hosts. Typical IDL compilers, ...

6 Modern languages and Microsoft's component object model

David N. Gray, John Hotchkiss, Seth LaForge, Andrew Shalit, Toby Weinberg

May 1998 **Communications of the ACM**, Volume 41 Issue 5

Full text available:  pdf(340.03 KB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#), [review](#)

7 Why IDLs are Not Ideal

A. Kaplan, J. Ridgway, J. C. Wileden



April 1998 **Proceedings of the 9th International Workshop on Software Specification and Design**

Full text available:  pdf(51.63 KB)  Additional Information: [full citation](#)
[Publisher Site](#)

8 Technical papers: software design: DADO: enhancing middleware to support crosscutting features in distributed, heterogeneous systems

Eric Wohlstadter, Stoney Jackson, Premkumar Devanbu

May 2003 **Proceedings of the 25th International Conference on Software Engineering**

Full text available:  pdf(1.56 MB)  Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)
[Publisher Site](#)

Some "non-" or "extra-functional" features, such as reliability, security, and tracing, defy modularization mechanisms in programming languages. This makes such features hard to design, implement, and maintain. Implementing such features within a single platform, using a single language, is hard enough. With distributed, heterogeneous (DH) systems, these features induce complex implementations which cross-cut different languages, OSs, and hardware platforms, while still needing to share data and ...


9 An extensible knowledge base management system for supporting rule-based interoperability among heterogeneous systems

Stanley Y. W. Su, Herman Lam, Javier Arroyo-Figueroa, Tsae-Feng Yu, Zhidong Yang
 December 1995 **Proceedings of the fourth international conference on Information and knowledge management**

Full text available:  pdf(1.17 MB) Additional Information: [full citation](#), [references](#), [index terms](#)

10 Developing language neutral class libraries with the System Object Model (SOM)

Mike Conner, Nurcan Coskun, Scott Danforth, Larry Loucks, Andy Martin, Larry Raper, Roger Sessions
 December 1992 **ACM SIGPLAN OOPS Messenger , Addendum to the proceedings on Object-oriented programming systems, languages, and applications (Addendum)**, Volume 4 Issue 2

Full text available:  pdf(360.02 KB) Additional Information: [full citation](#), [references](#), [index terms](#)

11 The direct cost of virtual function calls in C++

Karel Driesen, Urs Hölzle
 October 1996 **ACM SIGPLAN Notices , Proceedings of the 11th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications**, Volume 31 Issue 10

Full text available:  pdf(2.03 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

We study the direct cost of virtual function calls in C++ programs, assuming the standard implementation using virtual function tables. We measure this overhead experimentally for a number of large benchmark programs, using a combination of executable inspection and processor simulation. Our results show that the C++ programs measured spend a median of 5.2% of their time and 3.7% of their instructions in dispatch code. For "all virtuals" versions of the programs, the median overhead rises to 13. ...

12 Evaluating architectures for multithreaded object request brokers

Douglas C. Schmidt
 October 1998 **Communications of the ACM**, Volume 41 Issue 10

Full text available:  pdf(202.09 KB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)


13 Interoperability

Peter Wegner
 March 1996 **ACM Computing Surveys (CSUR)**, Volume 28 Issue 1

Full text available:  pdf(146.10 KB) Additional Information: [full citation](#), [references](#), [index terms](#)

14 Measuring the performance of communication middleware on high-speed networks

Aniruddha Gokhale, Douglas C. Schmidt

August 1996 **ACM SIGCOMM Computer Communication Review , Conference proceedings on Applications, technologies, architectures, and protocols for computer communications**, Volume 26 Issue 4Full text available:  [pdf\(270.13 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Conventional implementations of communication middleware (such as CORBA and traditional RPC toolkits) incur considerable over-head when used for performance-sensitive applications over high-speed networks. As gigabit networks become pervasive, inefficient middleware will force programmers to use lower-level mechanisms to achieve the necessary transfer rates. This is a serious problem for mission/life-critical applications (such as satellite surveillance and medical imaging). This paper compares t ...

15 Data integration and sharing II: Scientific data repositories: designing for a moving target

Etzard Stolte, Christoph von Praun, Gustavo Alonso, Thomas Gross

June 2003 **Proceedings of the 2003 ACM SIGMOD international conference on Management of data**Full text available:  [pdf\(739.27 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Managing scientific data warehouses requires constant adaptations to cope with changes in processing algorithms, computing environments, database schemas, and usage patterns. We have faced this challenge in the RHESSI Experimental Data Center (HEDC), a datacenter for the RHESSI NASA spacecraft. In this paper we describe our experience in developing HEDC and discuss in detail the design choices made. To successfully accommodate typical adaptations encountered in scientific data management systems ...

16 Java 2 distributed object middleware performance analysis and optimization

Matjaz B. Juric, Ivan Rozman, Simon Nash

August 2000 **ACM SIGPLAN Notices**, Volume 35 Issue 8Full text available:  [pdf\(1.31 MB\)](#) Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

This paper is focused on the performance analysis, comparison and optimization of distributed object middleware for Java 2: RMI (Remote Method Invocation), CORBA IDL (Interface Definition Language) and RMI-IIOP (Remote Method Invocation on Internet Inter-ORB Protocol). The paper presents the following contributions to the research on distributed object performance. First, a detailed performance analysis is provided with the comparison. These results help to understand how the models perform. Sec ...

Keywords: CORBA, IDL, IIOP, Java, RMI, performance analysis and optimization

17 CodeBricks: code fragments as building blocks

Giuseppe Attardi, Antonio Cisternino, Andrew Kennedy

June 2003 **ACM SIGPLAN Notices , Proceedings of the 2003 ACM SIGPLAN workshop on Partial evaluation and semantics-based program manipulation**, Volume 38 Issue 10Full text available:  [pdf\(294.34 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

We present a framework for code generation that allows programs to manipulate and generate code at the source level while the joining and splicing of executable code is carried out automatically at the intermediate code/VM level. The framework introduces a data type

Code to represent code fragments: methods/operators from this class are used to reify a method from a class, producing its representation as an object of type Code. Code objects can be combined by partial application to other Code ob ...

Keywords: domain specific language, generative programming, metaprogramming, multistage programming, program generation, program transformation, reflection

18 Web-based specification and integration of legacy services

Ying Zou, Kostas Kontogiannis

November 2000 **Proceedings of the 2000 conference of the Centre for Advanced Studies on Collaborative research**


Full text available:  [pdf\(279.28 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

With the explosive growth of the Internet, businesses of all sizes aim on applying networkwide solutions to their IT infrastructures, migrating their legacy business processes into web-based environments, and establishing their own on-line services. To facilitate process and service integration, a complete and information rich service description language, is essential for server processes to be specified and for client processes to be able to locate services that are available in Web-enabled re ...

19 Object-oriented technology: Using SOM for tool integration

Christina Lau

October 1993 **Proceedings of the 1993 conference of the Centre for Advanced Studies on Collaborative research: software engineering - Volume 1**

Full text available:  [pdf\(639.08 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#)

IBM's System Object Model (SOM) is an object-oriented programming interface for building binary class libraries. When changes are made to a SOM class, client programs that use the SOM class will not need to be recompiled. The SOM Toolkit provides a set of frameworks for building object-oriented applications. These include Distributed SOM, the Persistence Framework, the Replication Framework, and the Emitter Framework. In this paper, we discuss our experience using SOM to handle the control aspect ...

20 Quantifying aspects in middleware platforms

Charles Zhang, Hans-Arno. Jacobsen

March 2003 **Proceedings of the 2nd international conference on Aspect-oriented software development**

Full text available:  [pdf\(1.26 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Middleware technologies such as Web Services, CORBA and DCOM have been very successful in solving distributed computing problems for a large family of application domains. As middleware systems are getting widely adopted and more functionally mature, it is also increasingly difficult for the architecture of middleware to achieve a high level of adaptability and configurability, due to the limitations of traditional software decomposition methods. Aspect oriented programming has brought us new de ...

Keywords: aspect mining, aspect oriented programming, distributed systems, evaluation and metrics, middleware, software architecture

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)